INFINITE SOLUTIONS FROM INFINITE PERIPHERALS

PP-50

PDA Printer

Developer's Guide

INFINITE PERIPHERALS
PROVIDER OF CUSTOM RECEIPT PRINTING SOLUTIONS
ATLANTA • CHICAGO • DALLAS • LOS ANGELES • NEW YORK

Manufactured by Datecs

www.ipcprint.com

Contents

Technical Data *3*

Overview *4*

Compatible products: 4

Printer Control Methods *5*

Programming Card Reader *6*

Connecting External Devices *7*

External Devices Functions *8*

Palm Devices *9*

Metrowerks Code Warrior for Palm OS: 9

Falch.NET DeveloperStudio: 9

AppForge: 9

Satellite Forms: 9

Blackberry Devices *10*

Blackberry C++ SDK: 10

Blackberry Java SDK: 10

Pocket PC Devices *11*

PocketPC iPAQ Driver/SDK (ARM): 11

Direct Control Method *12*

PP-50 Resident Commands List 13

Commands Details 15

PP-50 Serial Port Pin Assignments *30*

PP-50 Carrier Pin Assignments *31*

Pin Assignments for the PP-50 Palm III carrier: 31

Pin Assignments for the PP-50 Palm V carrier: 31

Pin Assignments for the PP-50 Visor carrier: 31

Appendix A *32*
Card Reader Example Code: 32
Appendix B *33*
External Function Example Code: 33
Appendix C *34*
Serial Cable Examples: 34

Technical Data

Printing method	Line thermal dot printing
Printing speed	up to 50 mm per second
Dot density	8 dot/mm, 203 dpi – Horizontal and Vertical
Resident fonts	A – 12 x 24 B – 9 x 16
Loadable fonts	C – 12 x 24 D – 9 x 16
Logo Registration	1 Black & White BMP format (1-bit per pixel) Size: 384 x 240
Printing columns	Font A/C – 32 columns Font B/D – 42 columns
Serial Communications	Default: 57600, N , 8, 1 (Baud, Parity, Data Bits, Stop Bit) Using Hardware Handshaking
Power supply	Rechargeable battery pack, 7.2V @ 1.3, 1.5 or 1.8 AH Battery capacity: ~ 4 Rolls (Alpha-numeric data) AC/DC adapter 12V @ 0.5A
Environment	Operating temp. 0°C- 45°C @ 10 – 90 % RH Storage temp. -25°C - 70°C @ 10 – 90 % RH
Reliability	50 Million pulses 50 km
Dimensions	L – 195 mm, H – 50 mm, W – 85 mm Weight – 420 grams with battery and paper roll
Thermal paper	Roll 57 mm wide, 40 mm Ø (~ 80 feet)

Overview

The PP-50 portable thermal printer is designed specifically for use with the world's most popular palm-size computers. The following is a list of devices that the PP-50 currently supports.

Compatible products:

PP-50 III	Palm III, TRGPro, Handera 330, IBM WorkPad III, Symbol 1500
PP-50 V	Palm V, Palm Vx, IBM WorkPad V
PP-50 VII	Palm VII
PP-50 VIS	Visor Deluxe, Visor Pro, Visor Platinum, Visor Neo
PP-50 500	Palm 125, 130, 500, 505 and 515
PP-50 705	Palm 705, Tungsten W
PP-50 IPAQ	IPAQ (3800, 3900, and 5400 series)
PP-50 Treo	Treo 180 and 270
PP-50 B957	Blackberry 957
PP-50 B5810	Blackberry 5810

The PP-50 can be used in a variety of applications were hardcopy printouts and magnetic card reading is required.

✓ ***End Users –*** to print from standard palm applications – Date Book, Address, To Do and Memo, as well as freely resizable screen images generated by any PALM OS application or selected parts of it.

✓ ***Manufactures –*** for shipping, receiving goods, date and time printing, inventory control and W.I.P. management.

✓ ***Distributors –*** for marking prices, sales receipts, shelf labeling, route deliveries, pallets and packages labeling and portable POS.

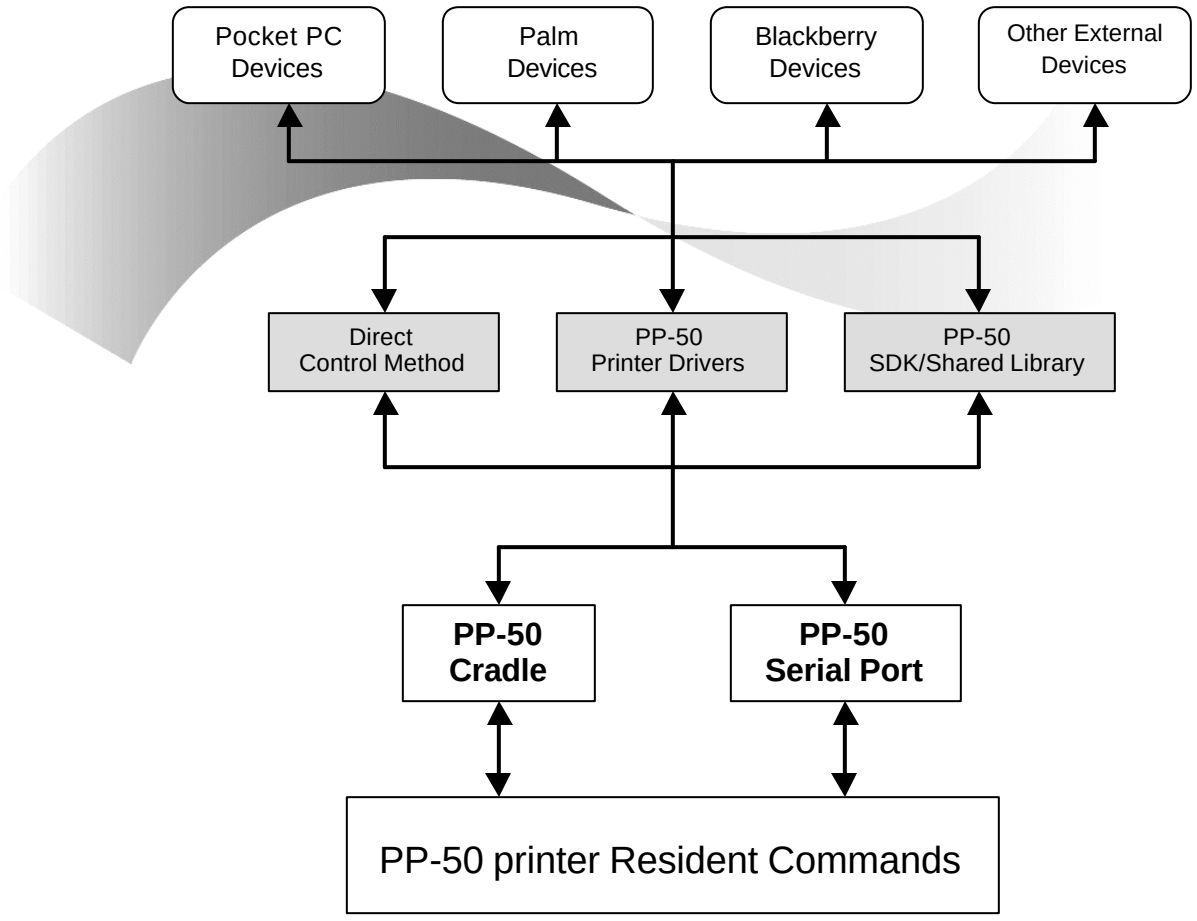
The PP-50 can also serve as a Hot Sync cradle and recharging station for palm devices with the use of an optional cable.

Printer Control Methods

There are three (3) possible methods to control the PP-50 thermal printer. As shown in the diagram below. Applications running on various palm size devices can be use to control the PP-50 using either the direct control method or using one of the available printer drivers or SDK (gray box).

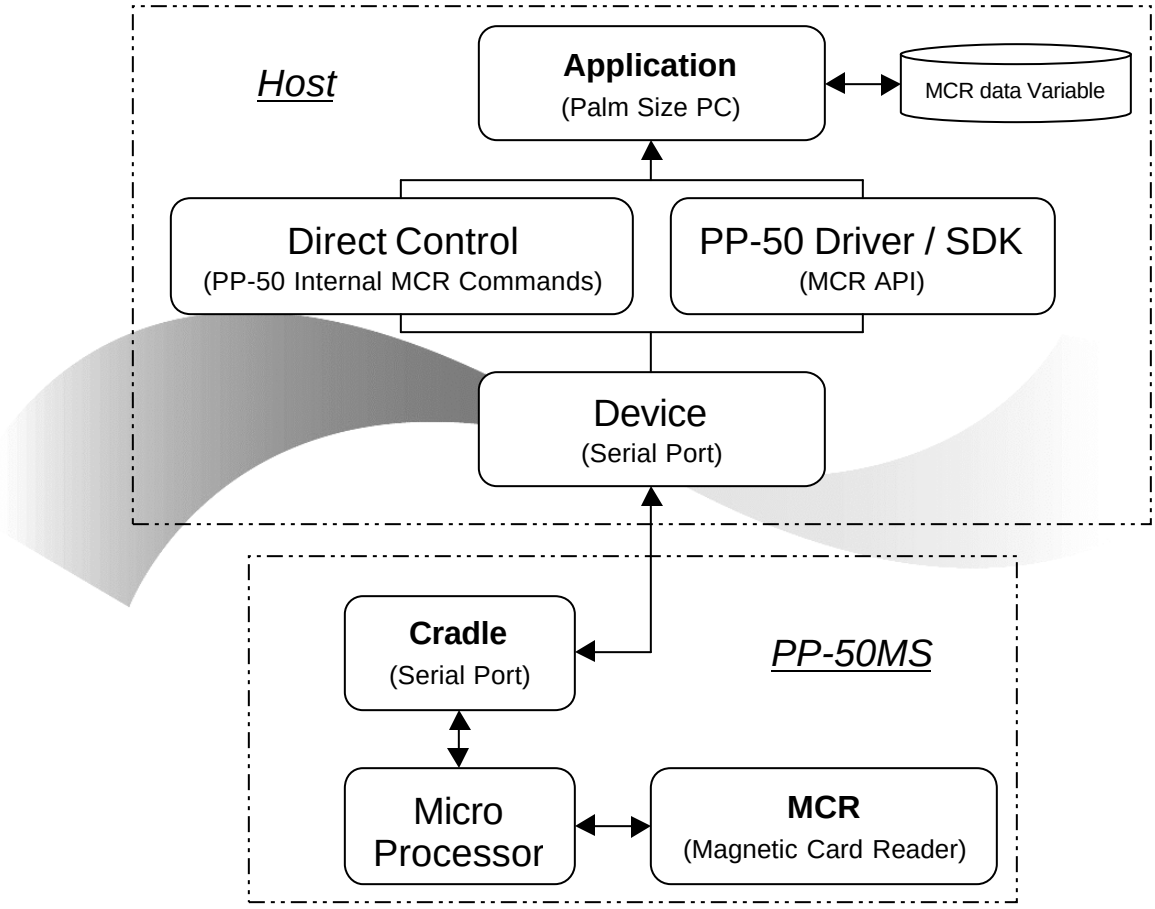
The control method use is limited by the Driver or SDK's compatibility with the various application development tools. See section on your device for details on Driver and SDK compatibility.

For example, if the development tool you are using is not compatible with the SDK, then controlling the PP-50 can be accomplish by the Direct Control method that access the PP-50 resident commands directly.



Programming Card Reader

In order to use the Magnetic Card Reader (MCR) an application must be written to activate the MCR and store the information read on the host device usually a PDA or palm size computer. The PP-50 Driver and SDKs provide a host of functions or APIs that simplify the task of activating the MCR and reading the data. The following is a simple block diagram showing how this is accomplish using the PP-50 Driver, SDKs, and Direct Control methods.



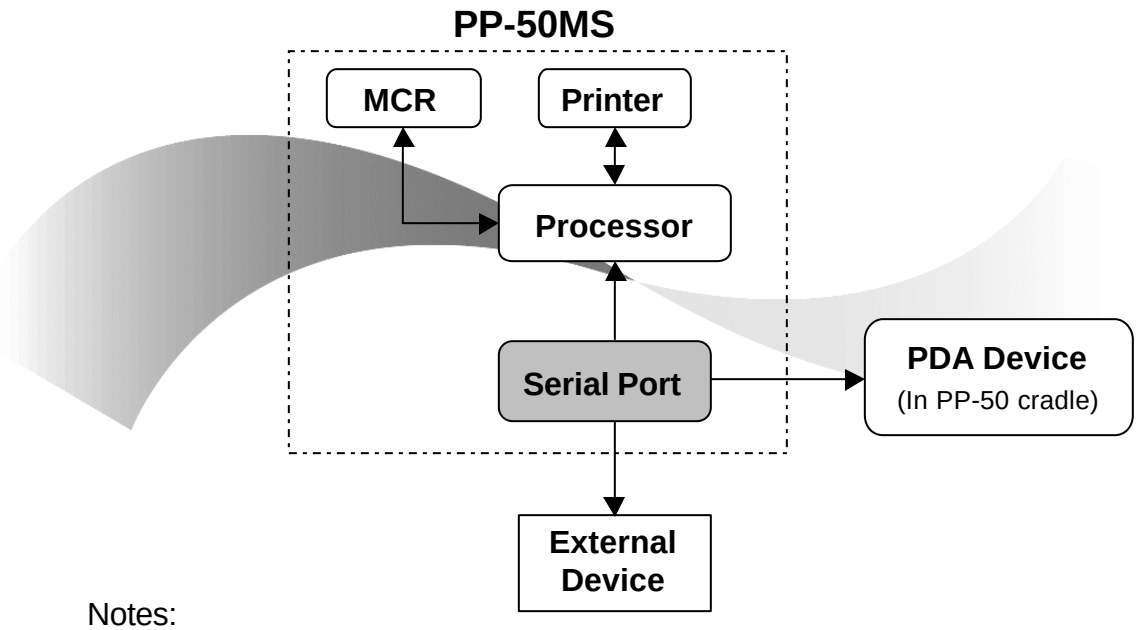
The application calls the Driver/SDK MCR API or sends the PP-50 internal MCR command depending on method used to communicate with the PP-50. Within the application a storage variable is defined and used to store returned data. The read data can then be process for further use by the applications.

See Driver/SDK documentation for programming for details.

Connecting External Devices

Connecting external devices to the PP-50 serial port expands the capability of the PP-50 printer. Bi-directional communications through the serial port can be use to connect Barcode readers and other external devices such as cell phones. In order to use the SDK's external functions developers must obtain a serial number and register it with the SDK from within their application. For details on obtaining a serial number, contact Infinite Peripherals, Inc. www.ipcprint.com.

The figure below is block diagram of the PP-50MS. A common serial port interfaces with both the PDA device and External device. Because of this configuration, special functions in the SDK are use to direct the flow of data so that it can be process correctly by the intended device.



Notes:

- Care must be taken when connecting external device to the PP-50 serial port as damage to port could result in the PP-50 not being able to communicate with the PDA.
- The PP-50 serial port does not implement all the signals from the RS232C Standards. *Refer to Serial Port Pin Assignment section for details*.
- Drivers and SDKs have limited support for external devices. External functions were added to most of the drivers and SDKs mentioned in this document however please review the Driver and SDK documentation for details.

External Devices Functions

The following example functions are use to communicate to external devices connected to the PP-50 serial port.

Before calling any of the DPSDK_Ext functions, the application must register the serial number that has been supplied to you by IPC. These external function can not be access without first registering using the DPSDK_REGISTER function.

DPSDK_REGISTER -

This function must be called before any other external communication. If the registration is unsuccessful, no external communication will be made.

DPSDK_ExtOpenConnection –

Initializes external connection with device attached to PP-50's port. The printer must be initialized successfully before. The function sleeps the printer, ends current connection and initializes a new one with the specified parameters.

DPSDK_ExtSendData –

Sends data to external device.

DPSDK_ExtReceiveData –

Receives data from external device.

DPSDK_ExtCloseConnection –

Closes the external connection, initializes a normal connection with PP-50 and awakes it.

Note: For more information on using the External Device functions, refer to the Reference.htm files included with the PP-50 Drivers and SDK.

Palm Devices

The SDK and Shared Library is designed to aid software engineers in the development of PALM OS applications to be used with the PP-50 thermal printer. The PP-50 Shared Library is compatible with Metrowerks Code Warrior, AppForge for VB, and Satellite Forms development tools with use of separate SDKs for the above mention IDE.

The following configuration is required to use the Printer SDK/Shared Library with the development systems mentioned above.

Metrowerks Code Warrior for Palm OS:

CodeWarior 8.0
CodeWarior 6.0

To use the PP-50 printer SDK/Shared Library with Code Warrior, the "DPSDKLib.prc" file must be loaded onto the Palm device. The "DPSDKLib.h" header file must be included in all Code Warrior projects.

Falch.NET DeveloperStudio:

Place the file "DPFN.o" in your library path or in the current directory and add in to the string of Linker options of the project, The file "DatecsPrinter.h" must be placed in the current directory or in the PalmSDK include directory!

AppForge:

To use Palm Printer SDK for AppForge, the "AppForgeSDK.prc" file must be loaded onto the Palm device. The "AppForgeSDK.bas" code module must be added to AppForge projects.

Satellite Forms:

To use Datecs PalmPrinter drivers for Satellite Forms the two files DP4SF.prc" and "DP4SF.inf" must be in Satellite Forms\extensions directory.

Note: For more information, a set of readme and Reference.html files is with the SDK of each of the mentioned IDE above.

Blackberry Devices

The Blackberry SDK and Shared Library is designed to aid software engineers in the development of C++ and Java OS applications to be used with the PP-50 thermal printer. The PP-50 Shared Library is compatible with Blackberry Handheld SDK 2.5.0 and Blackberry JDE.

The following configuration is required to use the Printer SDK/Shared Library with the development systems mentioned above.

Blackberry C++ SDK:

This SDK provides all required information about BlackBerry Library "GSDK.DLL" including sample code for quick and easy implementation in the different Integrated Development Environments (IDE) while developing your own application with printing ability.

Blackberry Java SDK:

This SDK provides all required information about BlackBerry Library "BBerrySDK.jar" including sample code for quick and easy implementation in the different Integrated Development Enviroments (IDE) while developing your own palm application with printing ability. To use this SDK please copy the file to your project's folder, then add it from the project properties and include the following line to the code: import sdk.BBerrySDK.*;

Note: For more information, a set of readme and Reference.html files is with the SDK of each of the mentioned IDE above.

Pocket PC Devices

Support for PocketPC devices come in the form of PocketPC (ARM) driver/SDK. This Driver/SDK aids software engineers in the development of PocketPC applications that can be used with the PP-50 thermal printer.

PocketPC iPAQ Driver/SDK (ARM):

The PocketPC driver follows the Window guideline for printer drivers with one exception. "Page size" – the page size it is calculated as proportional to given size, for example A4 size is treated as page with dimensions 48mm x 68 mm. User can obtain information about the driver via dc function GetDeviceCaps.

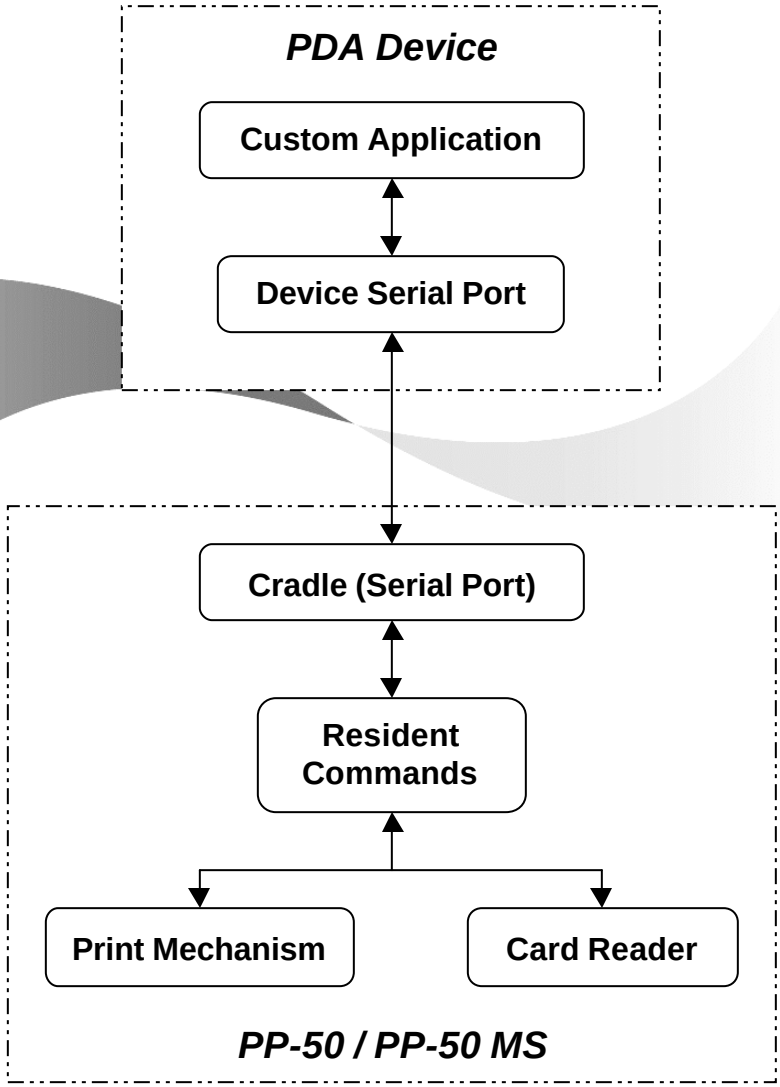
The driver incorporates functions such as PASSTHROUGH that enables the developers to send printer resident commands to the PP-50 through the driver. Other commands such as the READCARD enable applications to retrieve data from the magnetic card reader.

Note: For more information on using the PocketPC driver, refer to the ReferencePC.htm files included with the driver or SDK.

Direct Control Method

The PP-50 thermal printer has a resident command set that provides Text Formatting, Barcodes, and Graphics printing capability.

Using this method gives programmers greater flexibility and control of the PP-50 however, using this method also requires more programming efforts.



PP-50 Resident Commands List

The Direct Control and PASSTHROUGH method uses the “Escape” control sequence that starts with the ASCII code “ESC” or “GS” directly to the PP-50 printer via device serial port.

For example:

ASCII CODE:	ESC	E	1
DECIMAL:	27	69	01
HEXADECIMAL:	1B	45	01

OR

ASCII CODE:	GS	x	1
DECIMAL:	29	119	01
HEXADECIMAL:	1D	77	01

	Command	Description
1	HT	Horizontal tab command
2	LF	Printing and paper feed
3	CR	Ignored
4	ESC SP	Setting the space between characters
5	ESC$	Setting the absolute position for print
6	ESC%	Set/cancel custom selected characters for printing
7	ESC&	Loading custom selected symbols
8	ESC!	Setting the text printing mode
9	ESC*	Print graphic information
10	ESC -	Set/cancel underlining
11	ESC .	Diagnostic information /self-test/
12	ESC 2	Specifying 1/6-inch line height
13	ESC 3	Specifying n/203 inch line height
14	ESC =	Data input control
15	ESC?	Mag-stripe card reading
16	ESC @	Initializing the Printer
17	ESC D	Setting horizontal tab position
18	ESC E	Setting/canceling bold print
19	ESC G	Setting/canceling bold print
20	ESC J	Printing and feeding paper by n/203 inch
21	ESC S	Setting interface baud rate
22	ESC T	Printing short diagnostic information

Command List (Cont.)

	Command	Description
23	ESC V	Specifying/canceling 90° -right- turned characters
24	ESC Y	Set the print intensity
25	ESC \	Sst the relative horizontal printing position
26	ESC`	Returning data on battery voltage and printer head temperature
27	ESC a	Aligning the characters
28	ESC c5	Enabling/disabling the function of the LF button
29	ESC d	Printing and feeding the paper by n lines
30	ESC v	Transmitting the printer status
31	ESC x	Set the automatic switch-off time
32	ESC {	Setting/canceling the inverted characters
33	GS k	Printing the bar code
34	GS w	Set the bar-code horizontal size
35	GS h	Set the bar-code height
36	GS H	Set the position of the bar-code duplicate text
37	GS f	Set the font of the bar-code duplicate text
38	GS *	Loading a graphic image (logo)
39	GS /	Print the downloaded graphic image
40	GS :	Start/end of a macros definition
41	GS ^	Executing the macros
42	GS L	Left margin alignment setting
43	FS &	Enable the table of JIS symbols
44	FS .	Disable the JIS table of symbols
45	FS C	Enable/Disable Shift-JIS setup
46	FS !	Set attributes for printing in the JIS and Shift-JIS modes

Commands Details

HT
Horizontal Tab Command (HT)

[Code] **[09h]**
[Outline] Shifts the printing position to the next horizontal tab. The horizontal tab position is set by ESC D.

The default tab setting is at each 8 characters (9, 17, 25 etc.) of font “A”.

LF
Paper Feed Command (LF)

[Code] **[0Ah]**
Prints data from the buffer and feeds paper to the set line pitch.

CR
Print Line command

[Code] **[0Dh]**
This command is ignored.

ESC SP
Sets the space between the characters

[Code] **[1Bh] + [20h] + [n]**
[0 <= **n** <= 20h]
The entered data is binary. The distance between the characters is measured in dots (1/203 inches).

The initial value is **n=0**. When the width of characters is doubled, the distance between them is relatively doubled too.

ESC $
Setts the Absolute Position for print

[Code] **[1Bh] + [24h] + n1 + n2**
0 <= **n1** <= FF Horizontal shift in dots (LS byte)
0 <= **n2** <= 01 Horizontal shift in dots (MS byte)
The shift is **n1 + 256*n2** dots. After the end of the line no positions are acceptable.

Command Details (Cont.)

ESC %
Select/cancel custom selected characters for printing

[Code] **[1Bh] + [25h]+ n**
n may have a value between 0 and 255 but only the LS bit is of importance:
0 Loadable symbols are selected
1 The in-built font is selected

ESC &
Load custom selected symbols

[Code] [1Bh]+[26h] +a + n + m + D11 + … + D (m-n+1)k
20h <= n <=mFFh

a is the number of the sub-command, which can be:

0 or **'0' :** Copies the in-built font A over the loadable font A. All parameters after the number of the command are not entered.
1 or **'1'**: Copies the in-built font B over the loadable font B. All parameters after the number of the command are not entered.
2 or **'2':** Defines a set of consecutive symbols for font A (12 x 24).
3 or **'3':** Defines a set of consecutive symbols for font B (9 x 16).

n is the ASCII code of the first and **m** is the ASCII code of the last of (**m-n+1**) the consecutive symbols. For defining only one symbol **m=n.**

The data for the symbols is marked with **Dij.** Each symbol from font A is defined with 48 bytes and from font B – by 16 bytes.

The symbol from font A is set from left to right and downward, two bytes for each horizontal line where only the MS half of the second is used. Each bit sets one dot, units are in black and the start is marked with the most MS bit.
The symbols are stored after the printer is switched OFF.

Command Details (Cont.)

ESC !
Setting the text printing mode.

[Code] **[1Bh]+[21h] +n**
The entered data is binary.
Each bit of **n** has the following meaning:

Bit	Function	Value 0	Value 1
0	Character Font	Font A (12 x 24)	Font B (9 x 16)
1	Undefined		
2	Undefined		
3	Bold	Canceled	Set
4	Double height	Canceled	Set
5	Double width	Canceled	Set
6	Undefined		
7	Underline	Canceled	Set

The printed sign is underlined to its full width. The spaces, entered with the horizontal tabulator, are not underlined as well as those, rotated to 90°C.

ESC *
Print graphic information

[Code] **[1Bh] 2Ah] + m + n1 + n2 + D1+…+Dk**
m (0, 1, 20h or 21h) Graphic mode (see the table below).

0 <= **n1** <= FF Sets the number of horizontal dots (LS byte)
0 <= **n2** <= 01 Sets the number of horizontal dots (MS byte)
Di (**i** from 1 to **k**) Data on the graphic image

The number of horizontal dots is **n1+n2*256.**
The number of data bytes **k** is **n1 + 256*n2** for mode 0 and 1 and **(n1 + 256*n2)** for 20h and 21h.
The units in each data byte correspond to black dots.
Data is sent in vertical columns downward and from left to right, 1 or 3 bytes in a column depending on the selected mode.

Command Details (Cont.)

		Vertical Direction		Horizontal Direction	
m	**Mode**	**Dots**	**Dot Density**	**Dot Density**	**Max. Dots**
0	8-dot single density	8	67 DPI	101 DPI	192
1	8-dot double density	8	67 DPI	203 DPI	384
20h	24-dot single density	24	203 DPI	101 DPI	192
21h	24-dot double density	24	203 DPI	203 DPI	384

Given invalid value of **m** or **n2** the data is processed as symbols for printing.

The command has a second form with three new modes:
Code: **[1Bh] + [2Ah] + m + n + {a + [00h]} + D1 +…+ Dk**

Data on the graphic area is sent with a size of **n*8** horizontal dots and 24 vertical dots with or without data compressing depending on **m**. Both modes have a high density value (203 x 203 dots/inch).
m may be:

10h Non-compressed data 24 lines high. Byte **a** and byte **00h** are not sent.
11h Compressed data 24 lines high. Byte **a** and byte **00h** are not sent.
12h Compressed data with a height of **a** lines.

0<=**n**<=40h defines the horizontal size.

Di is the graphic data, the number of which is **n*24** bytes in mode 10h. The compressed data in mode 11h must produce the same number but only after decompression. The number of the bytes for mode 12h must be **a*n** (after decompression).

The compression in the 11h and 12h modes is similar to the PCX monochrome graphic format. If the two MS bits in the current byte are 1 then the rest define a repeat counter from 0 to 63 and the next byte will contain the repeated data. If at least one of the two MS bits is 0, the byte contains data and is directly used. If data for the printer contains a byte with the two most MS bits 1, it must be sent similar to the two bytes with counter 1.

The data for both modes is sent horizontally, from left to right and downward. Each byte contains 8 dots, units are black and the start is marked by the MS bit.

Command Details (Cont.)

ESC –
Set/Cancel Character Underline

[Code] **[1Bh] + [2Ah] + n**
The printed character is underlined to its full width with the exception of the part, skipped by the horizontal tab.
Inverted and rotated (to 90°) characters are not underlined.
The attached table shows the type of possible lines depending on the set value of **n**.

0 or **30h**	No underlining
1 or **31h**	Single thickness underline
2 or **32h**	Double thickness underline

ESC .
Printing diagnostic information (self-test).

[Code] **[1Bh] + [2Eh]**
A test page is printed containing current parameters including printing density, temperature of the printing head, battery level, mode (RS232 or IrDA) and baud rate in case RS232 is selected.

ESC 2
Setting 1/6-inch line height

[Code] **[1B] + [32h]**
If the line contains symbols the height of which cannot fit into the set size the line is automatically expanded to the required value.

ESC 3
Setting line height **n**/203 inches

[Code] **[1Bh] + [33h] +n**
n = 0 to 255
The default value is **n**=22h (1/6 inches)

Command Details (Cont.)

ESC =
Data Input Control

[Code] **[1Bh] + [3Dh] +n**
n may have a value between 0 to 255 but only the LS bit is of importance.
Value 0: The printer is selected
Value 1: The printer is not selected.
When the printer is not selected it does not receive data and the only command which is executed is ESC=n with the most LS bit 1.
The printer is selected by default.

ESC ?
Mag-stripe card reading

[Code] **[1Bh] +[3Fh] + n**
When the command is received, the LED illuminates in red and the printer expects the card to be swiped through the reader. If after 10 seconds no card has been swiped, the command is automatically deactivated. The printer returns the contents of the tracks which have been read ending with 00h. If nothing has been read only 00h is returned.
n is a parameter defining the tracks we wish to read. For a three-track reader the possible values are:

bit 1: if set, reads Track 1
bit 2: if set, reads Track 2
bit 3: if set, reads Track 3
bit 4: if set, a “prefix” byte is added before the every track

Setting bit 4 - the tracks are prefixed with a single byte for better recognition of where they start. If other 3 bits are set to 1 all the 3 tracks read. For example ESC ? 7(number 7 representing the first 3 bits on) will read all the 3 tracks but will not divide the tracks with some prefix, you are still able to get the contents tho. While ESC ? 15(all 4 bits on) will read the 3 tracks and will return to you the information divided by some prefix byte.

ESC @
Initializing the printer

[Code] **[1Bh] + [40h]**
Clears the printing data from the buffer. The printer switches to its default settings (similar to when switching the printer ON).
The data in the serial buffer is not cleared.

ESC D
Setting the Horizontal Tab Positions

[Code] **[1Bh] + [44h] + n1+…+ nk + [00h]**

ni has a value from 0 to 255.
ni indicates the sequence number of the column counted from the beginning of the line minus 1. In order to define tab position in column 9 we will have to enter 8.

The length of the tab is equal to the width of the character multiplied by the digit **ni** (which is defined by this command) from the beginning of the line. The width of the character at this point includes the distance between the characters and is doubled when double increase is set.

The maximum number of tab positions is 32.

ESC D [00h] clears the set tab positions. After clearing, the horizontal tabulator is ignored.

ESC E
Set/cancel bold print

[Code] **[1Bh] + [45h] + n**
n may have a value between 0 to 255 but only the LS bit is of importance.
Value 0: Bold type is activated
Value 1: Bold type is canceled.
The command is valid only for font A (12 x 24)

ESC G
Set/cancel bold print (ESC G n)

[Code] **[1Bh] + [47h] + n**
The command is equivalent to ESC E

Command Details (Cont.)

ESC J
Print and feed paper by **n**/203 inch

[Code] **[1Bh] + [4Ah] + n**
n may have a value from 0 to 255.
Prints the data, accumulated in the graphic buffer, and advances paper to **n**/203 inches.

The set shifting is valid only for the current command.
The beginning of the line is accepted as a new starting point for printing.

ESC S
Setting the interface transmit speed (baud rate)

[Code] **[1Bh] + [53h] +n**
Sets the new baud rate value for the serial interface. The command is not executed in the IrDA mode. Possible values of the parameter **n** are:
0 or **'0'**: 2400 bps
1 or **'1'**: 9600 bps
2 or **'2'**: 19200 bps
3 or **'3'**: 57600 bps
4 or **'4'**: 115200 bps

The set baud rate is valid only when controlling the printer via a serial cable. The last set value remains valid after printer switch-OFF even when it has not been operated in the IrDA mode.
The default value is 1 (9600 bps).

ESC T
Printing short diagnostic information

[Code] **[1Bh] + [54h]**

ESC V
Setting/Canceling 90º rotated symbols
[Code] **[1Bh] + [56h] + n**
n may have a value between 0 to 255 but only the LS bit is of importance.
0 Cancel 90º rotation of symbols.
1 Set to 90º rotation of symbols.
Rotated symbols are not underlined.

Command Details (Cont.)

ESC Y
Selecting the print intensity level

[Code] **[1Bh] + [59h] + n**

n is between 0 and 5 or between '0' and '5' including:

0 or **'0'**	Set intensity 70 %
1 or **'1'**	Set intensity 80 %
2 or **'2'**	Set intensity 90 %
3 or **'3'**	Set intensity 100 %
4 or **'4'**	Set intensity 120 %
5 or **'5'**	Set intensity 150 %

Higher intensity printing may lead to lower printing speeds.
The default value is **3** (100%).

**ESC **
Specifying the relative horizontal printing position

[Code] **[1Bh] + [5Ch] + n1 + n2**

0 <= n1 <=FFh Horizontal shift in dots (LS byte)
0 <=n2 <=FFh Horizontal shift in dots (MS byte)
The shift is **n1 + 256*n2** dots. Positions before and after the line are unacceptable.

Shifting to the left of the current position is done by entering the addition to the necessary digit up to 65536 (N.=65536 – N).

ESC `
Returning data on battery voltage and printer head temperature

[Code] **[1Bh] + [60h]**
Two bytes of information are returned – the first is the voltage in tenths of a volt plus 20h while the second is the temperature of the printer head in Celsius plus 20h.

Command Details (Cont.)

ESC a
Alignment

[Code] **[1Bh] + [61h] + n**
n has a value between 0 and 2 or between '0' and '2'.
0 or **'0'** Left alignment
1 or **'1'** Center aligned
2 or **'2'** Right aligned
Default value is "0".

ESC c5
Enabling/Disabling The function of the LF button

[Code] **[1Bh] + [63h] + [35h] +n**
n may have a value between 0 to 255 but only the LS bit is of importance.
Value **0**: The LF button is enabled.
Value **1:** The LF button is disabled.
The default value is "0".

ESC d
Printing and Advancing paper by **n** lines

[Code] **[1Bh] + [64h] + n**
n may have a value between 0 to 255.
The data, accumulated in the graphic buffer is printed out and paper is advanced to **n** lines.
The beginning of the line is accepted as a new starting position for print.

ESC v
Receiving the printer status

[Code] **[1Bh] + [76h]**
The printer returns one byte in which only bit 2 is defined. Its meaning is:
Value 0: There is paper
Value 1: No paper

Command Details (Cont.)

ESC x
Setting the automatic Switch-OFF time

[Code] **[1Bh] + [78h] + n**
The command sets the time duration, after which the printer will be automatically switched off if no commands are sent, there is no IrDA connection and the button LF has not been pressed.
n is one byte with a value between 0 and 60 including and sets the time in minutes. When the value is 0 there is no automatic switch OFF. The set time is stored even after the printer is switched OFF.

The default value is 10 minutes.

ESC {
Set/Cancel Inverted Characters (180°)

[Code] **[1Bh] + [7Bh] + n**
n may have a value between 0 to 255 but only the LS bit is of importance.
0 Canceling inverted characters.
1 Setting inverted characters.
The default value is "0".

GS k
Printing Bar Codes

[Code] **[1Dh] + [6Bh] + n + Di + [00h]**
n sets the type of bar-code and may be:

n	Bar-code type
0	UPC-A
1	UPC-E
2	JAN13 (EAN)
3	JAN 8 (EAN)
4	CODE 39
5	ITF
6	CODABAR (NW-7)
7	CODE 128
8	CODE 93

Di designates the bar-code data. The necessary number and acceptable symbols depend on the type of the bar-code selected.

Command Details (Cont.)

GS w
Selecting the horizontal size of the bar code

[Code] **[1Dh] + [68h] + n**
n has a value between 2 and 4 and designates the thickness of one bar within the bar-code.

The default value of **n** is "3".

GS h
Selecting the height of the Bar Code

[Code] **[1Dh] + [68h] + n**
n has a value between 1 and FFh and sets the height of the bar-code in dots (1/203 inches).

The default value of **n** is 162.

GS h
Set the Printing Position of the bar-code duplicate text

[Code] **[1Dh] + [48h] + n**
n has a value between 0 and 3 or '0' or '3' included and sets the exact location of the duplicate text:

Value	Duplicate Text
0	No Printing
1	Above the bar code
2	Below the bar code
3	Both above and below the bar code

GS f
Sets the font of the bar-code duplicate text
[Code] **[1Dh] + [66h] + n**
n may have the following values:
0 Font A
1 Font B

Command Details (Cont.)

GS *
Loading a graphic image (logo) (GS*)

[Code] **[1Dh] + [2Ah] + n1 + n2 + Di+…+Dn**
n1 has a value between 1 and 127 and sets the horizontal size of the image.
N2 has a value between 1 and 248 and sets the vertical size of the image.
Di is the graphic image data. The data consists of **n1*n2** bytes from left to right and downward – **n1** bytes in each horizontal line (n1*8 dots) and **n2** lines. Each bit sets one dot – 1 corresponds to black. The total number of bytes cannot be greater than 16kB.

The command defines a graphic image made up from a number of dots defined with **n1** and **n2.** The image is stored after switch-OFF.
The loaded image is printed out with the command **GS/.**

GS /
Printing a downloaded graphic image

[Code] **[1Dh] + [2Fh] + m**
m sets the printing mode which may be:

m	Mode Name	Dots in Vertical Direction	Dots in Horizontal Direction
0	Normal mode	203 DPI	203 DPI
1	Double wide mode	203 DPI	101 DPI
2	Double high mode	101 DPI	203 DPI
3	DW / DH mode	101 DPI	101 DPI

When no graphic image has been loaded the command is ignored.
If sizes are greater than acceptable, the redundant part is not printed.

GS :
Start/End of a Macros Definition

[Code] **[1Dh] + [3Ah]**
Sets the start/end of a macros. Not more than 2048 bytes can be defined as macros. After the last data byte the command is sent once again to mark the end.

The macros is not deleted even after the execution of **ESC**@ (printer initialization). For this reason ESC@ may be included in it.
The printer is able to print during the definition of a macros.

Command Details (Cont.)

GS ^

Executing a Macros

[Code] **[1Dh] + [5Eh] + n1 + n2 + n3**

n1 has a value between 1 and 255: the number of times the macros has been executed.

n2 has a value between 1 and 255: a time interval between the execution of the macros in units of 100 milliseconds.

n3 – macros execution mode. Possible values:

0 Execution over a time interval set by **n2**

1 For every execution the **LF** button must be pressed.

GS L

Left margin alignment setting

[Code] **[1Dh] + [4Ch] + n1 + n2**

Sets the position in dots (1/203 inches for the beginning of each printed line. The command is executed only when sent at the beginning of the line. The set shift is **n1 + 256*n2** from left to right.

The default value is 0.

FS &

Enable the table of symbols JIS (FS &)

[Code] **[1Ch] + [26h]**

FS.

Disable the table of symbols JIS (FS .)

[Code] **[1Ch] + [2Eh]**

FS C

Enable/ Disable the Shift-JIS (FS C) setting

[Code] **[1Ch] + [43h] + n**

n may have the following values:

0 or **'0'**: The Shift-JIS is not been enabled

1 or **'1'**: The setting Shift-JIS is enabled

Command Details (Cont.)

FS !
Set attributes for printing in the JIS and Shift-JIS modes
[Code] **[1Ch] + [21h] + n**

n has a value between 00h and FFh, where the separate bits have the following meanings:

Bit	Function	Value 0	Value 1
0	Undefined		
1	Undefined		
2	Double width	Canceled	Enabled
3	Double height	Canceled	Enabled
4	Undefined		
5	Undefined		
6	Underline		
7	Underline	Canceled	Enabled

PP-50 Serial Port Pin Assignments

Connecting to an external device through the PP-50 serial port requires the use of custom cables *not supplied by Infinite Peripherals*. To aid in the building of these special cables the following is the PP-50's serial port pin assignment.

Pin	Name	Description
1	GND	
2	If connected to GND	Enable the HotSync and disable printer function
3	If connected to GND	Enable Vout on the pin 12
4	NC	
5	RTS	
6	NC	
7	TxD	For Printer
8	NC	
9	TxD (for PDA) or RxD (for Printer)	
10	NC	
11	CTS	
12	Vout	Direct battery or 5V regulated (<80mA), switched with a jumper on the interface board of PP-50 (only applies to new boards)
13	RxD	For PDA
14	-	
15	If connected to pin 14	Convert the ON/Feed button to HotSync button

PP-50 Carrier Pin Assignments

Pin Assignments for the PP-50 Palm III carrier:

Pin Number	Signal
1	NC
2	VBAT
3	RD
4	RTS
5	TD
6	CTS
7	Hot Sync
8	NC
9	NC
10	GND

Pin Assignments for the PP-50 Palm V carrier:

Pin Number	Signal
1	NC
2	VBAT
3	RD
4	RTS
5	TD
6	CTS
7	Hot Sync
8	V Charge
9	NC
10	GND

Pin Assignments for the PP-50 Visor carrier:

Pin Number	Signal
1	RD (TTL)
2	GND
3	Hot Sync
4	GND
5	NC
6	NC
7	NC
8	TD (TTL)

Appendix A

Card Reader Example Code:

The example code uses the Basic language to read credit card information from the PP-50.

```
Dim strSend as String
Dim intCount as Integer
Dim received as String          '---Reveive buffer
Dim datastream as String        '---Data buffer
Dim err as Integer
Dim delay as Integer            '---Time out delay
Dim cardTrack as Integer        '---Store tracks to read

'---Initialize buffer, counter and track to read

        receive = “”
        datastream = ""
        intCount = 1
        cardTrack = 0
        delay = 1

'---Send the read card information command

        Call SerialOpen(57600,0)                        '---Open the serial port
        strSend = chr(27)+chr(63)+chr(cardTrack)        '---Send the read card command
        err = SerialSend (strSend, 3)                   '---Send the command (3) bytes

'---Get the card information from the serial buffer one character at a time.

        err=SerialReceive (received,1, delay)
        Do while err = 0 And intCount <= 400
                datastream = datastream + received      '---Put the character read into buffer
                err = SerialReceive (received, 1, delay)
                intCount = intCount + 1
        Loop
```

Note: After successfully reading the data form the PP-50 into a buffer, field information must then be extracted using programming language string functions.

Appendix B

External Function Example Code:

The example code uses the C/C++ language to register the serial number with the SDK.

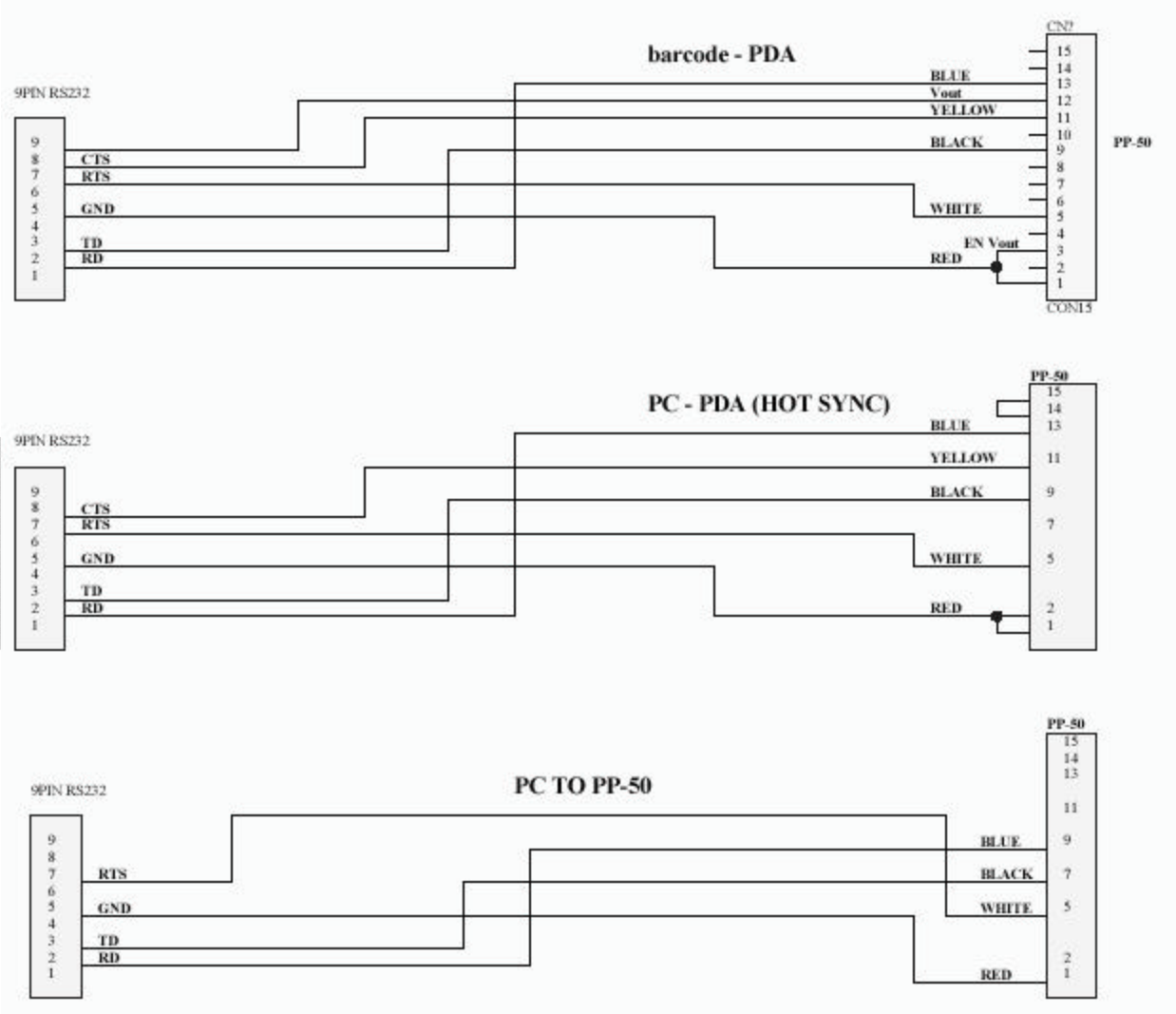
```
error = DPSDKLib_OpenLibrary(& DPSDK_Ref,&Context);

FatalErr(error != errNone, "Can't load DPSDK");

if(!DPSDK_REGISTER(DPSDK_Ref,"REGISTRATION NUMBER GOES HERE!!!")

FrmCustomAlert(InfoAlert,"cannot register !!!", 0,0);

if(DPSDK_InitPrinter(DPSDK_Ref))
        {
        if(DPSDK_ExtOpenConnection(DPSDK_Ref, 9600, 8, 1, 0)
                {
                        //send.... receive...
                        DPSDK_ExtCloseConnection(DPSDK_Ref);
                }
        }
else
        {
        FrmCustomAlert(InfoAlert,"cannot initialize printer !!!", 0,0);
        error = DPSDKLib_CloseLibrary(DPSDK_Ref,Context);
        }
```

Appendix C

Serial Cable Examples: